Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем
Параллельное программирование

Голиков Андрей Владиславович

Распараллеливание метода декомпозиции области типа Дирихле-Дирихле

Выпускная квалификационная работа

Научный руководитель:
д. ф.-м. н., проф. Корнеев В. Г.

Рецензент:
д. ф.-м. н., проф. Рябов В. М.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Parallel programming

Andrei Golikov

Parallelisation of the Dirichlet-Dirichlet domain decomposition method

Graduation Project

Scientific supervisor:
professor Vadim Korneev

Reviewer:
professor Victor Ryabov

Saint-Petersburg
2017

Оглавление

Введение

Для современных физических задач часто необходимо быстро решать уравнения в частных производных. При этом физические задачи могут характеризоваться абсолютно разными уравнениями на разных областях. Хотелось бы иметь метод, позволяющий на параллельной машине быстро решать подобные задачи, также учитывающий некоторые физические ее особенности, например, разнородность рассматриваемого материала.

В настоящее время наиболее популярными методами для численного решения уравнений в частных производных являются метод конечных разностей или метод конечных элементов. Таким образом, решение уравнения в частных производных сводится к решению большой алгебраической системы уравнений. Размер системы зависит от плотности сетки. Столь плотные сетки (порядка миллиона узлов и больше) оказываются необходимы для получения достаточно точных физических параметров рассматриваемого объекта, например, напряжения. Эти системы оказываются очень разреженными, и для их решения могут применяться различные методы, такие как метод LU-разложения, метод Якоби, метод Гаусса-Зейделя, варианты метода сопряженных градиентов. Также хочется иметь достаточно быстрый метод для решения больших СЛАУ, так как могут ставиться задачи мониторинга некоторой физической системы в живом времени. Однако указанные выше методы являются недостаточно хорошо гранулируемыми, поэтому отдельно хотелось бы выделить многосеточный метод, показывающий лучшую асимптотику на многопроцессорных системах.

Особенно хочется отметить метод декомпозиции области, использующийся для решения указанных систем линейных алгебраических уравнений и имеющий хорошие теоретические оценки времени исполнения. Он может включать в качестве своей части различные методы, указанные выше. В данной работе исследуются его оптимальные параметры при различных входных данных.

1. Постановка задачи

Исследовать различные варианты метода конечных элементов для решения уравнения Пуассона. Исследовать метод декомпозиции области типа Дирихле-Дирихле [8]. Реализовать параллельную программу, использующую этот метод. Применить программу для решения уравнения Пуассона на области вида рис. 1. Найти оптимальное число подобластей путем экспериментальных запусков на вычислительном кластере.

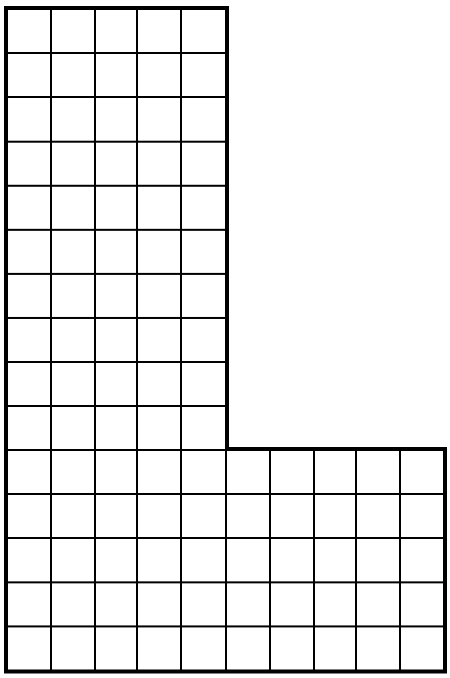

Рис. 1: Область задачи

2. Обзор литературы

Как методу конечных элементов, так и методу декомпозиции посвящено множество работ.

Одним из основных трудов на русский язык на текущее время является учебник Даутова и Карчевского [9]. Он содержит описание, способы построения и оценки для схем метода конечных элементов.

Также можно отметить пособие авторов Дугласа, Хаазе и Лангера, в котором рассмаривается большая часть вопросов, связанных с решением эллиптических уравнений в частных производных и их параллелизация, хотя и часть важных моментов, связанных с непосредственной реализацией всё-таки остаётся за его пределами [2].

Сходную тематику имеет и сборник статей под редакцией Брюасета и Твайто [1]. В нем освещено много вопросов численного решения уравнений в частных производных, в частности, есть статьи, посвященные многосеточному методу и методу декомпозиции области.

По методу декомпозиции одной из последних больших работ является книга Корнеева и Лангера, в которой рассматриваются дискретизации для двух- и трехмерных эллиптических уравнений в частных производных и строятся оценки для их сходимости [8].

Для реализации метода декомпозиции были использованы технология OpenMP [4], реализованная в компиляторе Intel Cluster Studio [11]. Также для быстрого преобразования Фурье была использована библиотека The Fastest Fourier Transform in the West [7], а реализация многосеточного метода взята из библиотеки AMGCL [3].

3. Метод конечных элементов

Для решения уравнений в частных производных зачастую используется метод конечных элементов. Этот метод позволяет получить численное решение эллиптических уравнений. Обоснование данного метода можно найти, например, в [9]. В простейших случаях, таких, как рассматриваемая задача, он совпадает с конечно-разностным методом.

В данной работе будет рассмотрено решение задачи Дирихле для уравнения Пуассона в области $\Omega \subset \mathbb{R}^2$. Задача формулируется как

$$\begin{cases} -\Delta u(x) = f(x), x \in \Omega \\ u(x)|_{\partial\Omega} = 0 \end{cases} \tag{3.1}$$

Для обоснования метода испольуется вариационный принцип. Слабая постановка задачи заключается в нахождении функции $u \subset \mathbb{M} = \mathbb{H}_0^1(\Omega)$ такой, что выполняется равенство

$$a(u, v) = (f, v), \forall v \in \mathbb{V}_0 \tag{3.2}$$

При этом пространство тестовых функций $\mathbb{V}_0$ можно выбирать разными способами, а билинейная форма $a(\cdot, \cdot) : \mathbb{M} \times \mathbb{V}_0 \to \mathbb{R}$ и линейная форма $(f, \cdot) : \mathbb{V}_0 \to \mathbb{R}$ определяются как

$$a(u, v) = \int_\Omega \nabla u \nabla v dx \tag{3.3}$$

$$(f, v) = \int_\Omega f(x)v(x)dx \tag{3.4}$$

При использовании метода Галёркина выбираются конечномерные подпространства $\mathbb{M}_h$ и $\mathbb{V}_h$ пространств $\mathbb{M}$ и $\mathbb{V}$. Тогда решение в методе Галёркина удовлетворяет равенству

$$a(u_h, v_h) = (f_h, v_h), \forall v_h \in \mathbb{V}_{0,h} \tag{3.5}$$

Далее опишем базис пространства функций, в котором мы будем работать.

В данной работе будут рассматриваться только области Ω с кусочнолинейной границей. Более того, все границы будут параллельны осям координат. Для таких областей можно задать разбиение области Ω на конечные элементы таким образом, чтобы элементы не имели общих внутренних точек и объединение их замыканий апроксимировало область задачи. В случае кусочно-линейных границ это объединение будет совпадать с ней. Такое разбиение называется триангуляцией. В нашем случае удобно взять одинаковые треугольные элементы.

В качестве базисных функций возьмем так называемые базисные функции Куранта (см. рис. 2). Это функции, задающиеся в узлах триангуляции. Они линейны на каждом элементе, прилегающем к данной вершине, равны 1 в вершине и нулю на остальной области (см. рис. 2). [2, 9]

После этого, проинегрировав первое уравнение в (3.1) по частям, получим

$$\int_\Omega \left(\frac{\partial u}{\partial x_1}\frac{\partial v}{\partial x_1} + \frac{\partial u}{\partial x_2}\frac{\partial v}{\partial x_2}\right)dx = \int_\Omega fvdx. \qquad (3.6)$$

Будем искать решение в пространстве, порожденном базисными элементами Куранта. Любая функция из этого пространства будет непрерывна на Ω, равна нулю на $\overline{\Omega}$ и непрерывна на каждом отдельном элементе. Тогда u^h представляется в виде

$$u^h(x) = \sum_{k,l=0}^{n} u^h_{kl}\varphi_{kl}(x), \qquad (3.7)$$

где $u^h_{kl} = u^h(x_{kl})$, а $\varphi_{kl}(x)$ – определенная выше базисная функция («шапочка Куранта»).

Построим систему уравнений для определения коэффициентов u^h_{kl}. Для этого посчитаем коэффициенты и правую часть в системе метода Галёркина [9]

$$a_{klk'l'} = \int_\Omega \left(\frac{\partial \varphi_{kl}}{\partial x_1}\frac{\partial \varphi_{k'l'}}{\partial x_1} + \frac{\partial \varphi_{kl}}{\partial x_2}\frac{\partial \varphi_{k'l'}}{\partial x_2}\right)dx, \quad b_{kl} = \int_\Omega f\varphi_{kl}dx. \qquad (3.8)$$

Понятно, что $a_{klk'l'}$ может быть не нулем только в том случае, когда пересечение областей Ω_{kl} , $\Omega_{k'l'}$ не пусто. Для фиксированных k, l та-

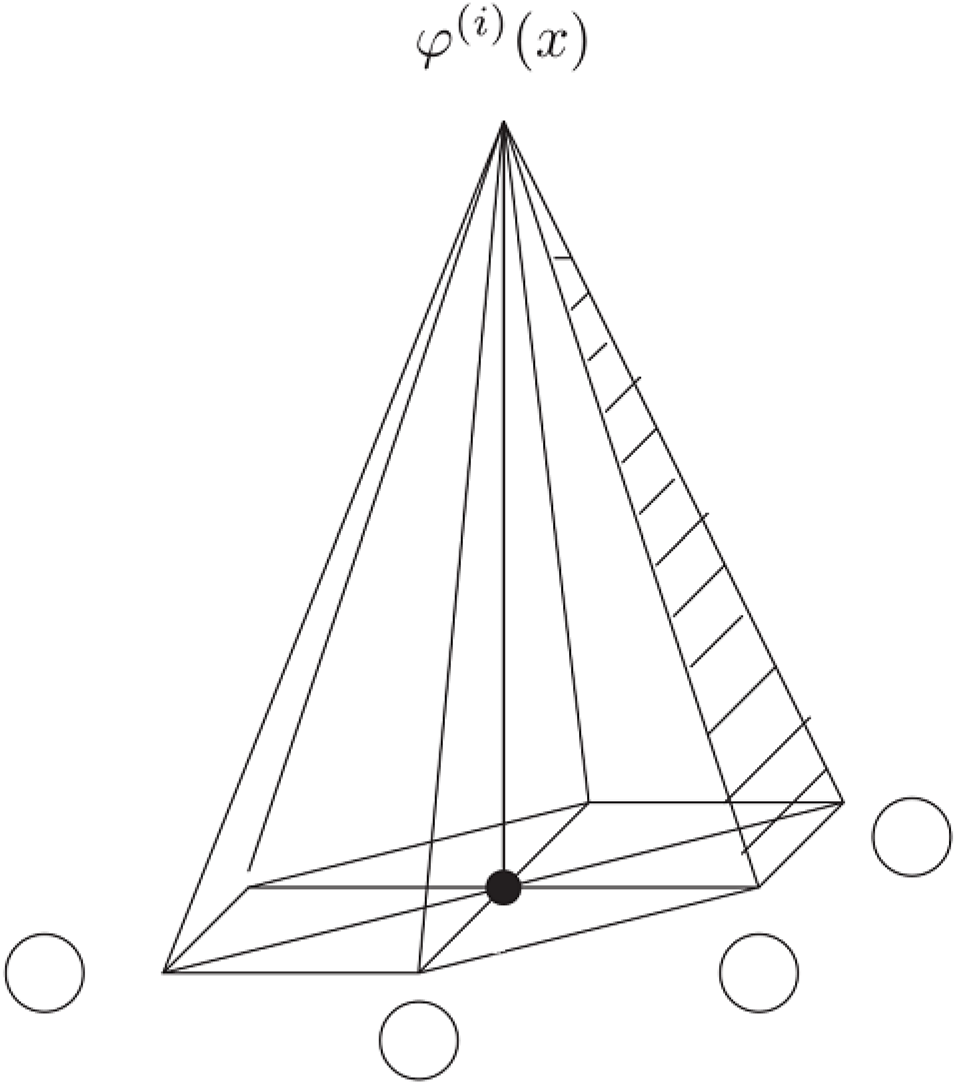

Рис. 2: Базисный элемент Куранта

ких областей Ω_{kl} как максимум шесть, а именно: $\Omega_{k-1,l}$, $\Omega_{k-1,l-1}$, $\Omega_{k,l-1}$, $\Omega_{k+1,l}$, $\Omega_{k+1,l+1}$, $\Omega_{k,l+1}$, $\Omega_{k,l+1}$. При вычислении коэффициентов $a_{klk'l'}$ потребуются значения производных функции φ_{kl} . Пронумеруем элементы, принадлежащие Ω_{kl}, как показано на рис. 3. Сведем результаты очевидных вычислений в таблицу 1 (n — номер треугольника). Теперь ясно, что

$$a_{k,l,k-1,l-1} = \int_{e_6 \cup e_1} \left(\frac{\partial \varphi_{kl}}{\partial x_1} \frac{\partial \varphi_{k-1,l-1}}{\partial x_1} + \frac{\partial \varphi_{kl}}{\partial x_2} \frac{\partial \varphi_{k-1,l-1}}{\partial x_2}\right) dx = 0,$$

$$a_{k,l,k-1,l-1} = 0, \; a_{k,l,k-1,l-1} = 0, \; a_{k,l,k-1,l-1} = -1$$

$$a_{k,l,k-1,l-1} = -1, \; a_{k,l,k-1,l-1} = -1, \; a_{k,l,k-1,l-1} = 4$$

n	$\partial\varphi_{kl}/\partial x_1$	$\partial\varphi_{kl}/\partial x_2$
1	0	$1/h$
2	$-1/h$	$1/h$
3	$-1/h$	0
4	0	$-1/h$
5	$1/h$	$-1/h$
6	$1/h$	0

Таблица 1: Производные функции φ_{kl}.

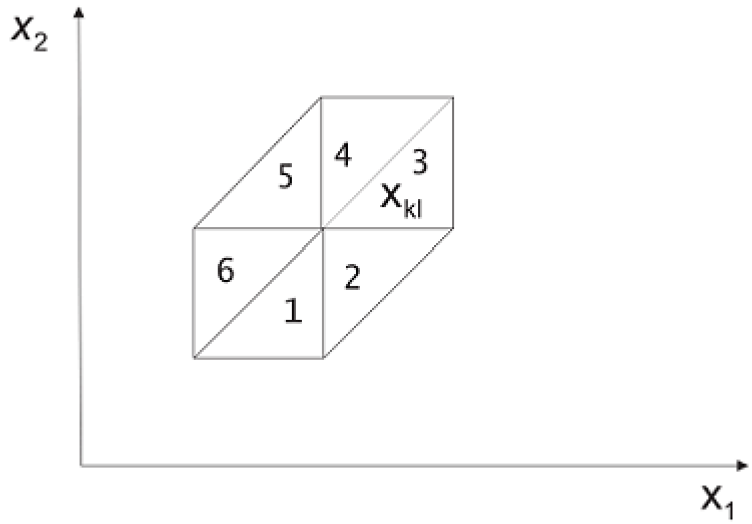

Рис. 3: Область Ω_{kl}; нумерация элементов

Это означает, что уравнение с номером k, l системы метода Галеркина принимает вид

$$-u^h_{k-1,l} - u^h_{k+1,l} + 4u^h_{k,l} - u^h_{k,l-1} - u^h_{k,l+1} = b_{kl}. \tag{3.9}$$

Заметим, что при достаточно малом h

$$b_{kl} = \int_{\Omega_{kl}} f(x)\varphi_{kl}(x)dx \approx f(x_{kl}) \int_{\Omega_{kl}} \varphi_{kl}(x)dx = f(x_{kl})h^2.$$

Записывая уравнения (3.9) во внутренних точках сетки L-образной области, и присоединяя к ним граничные условия, соответствующие (3.1), получим полную систему линейных алгебраических уравнений для отыскания приближенного решения в точках сетки.

4. Многосеточный метод

Для решения систем, получающихся в результате применения метода конечных элементов, зачастую применяют многосеточный метод. Это рекурсивный метод, основанный на использовании последовательности вложенных сеток и операторов перехода от одной сетки к другой и, при некоторых ограничениях на задачу, может применяться для решения произвольных систем линейных алгебраических уравнений. В случае его применения к эллиптическим уравнениям операторы перехода с одной сетки на другую могут задаваться исходя из параметров физической задачи.

В алгоритме 1 указан многосеточный метод без предварительного и пост-сглаживания, описанный в немного другом виде в пособии Крэйга, Хаазе и Лангера [2]. Для написания параллельной программы, реализующей данный метод, необходимо по крайней мере описать операторы ограничения(проекции), продолжения(интерполяции) и оператор точного решения на самой грубой сетке. В алгоритме также требуется указать константу γ, показывающую сколько раз применять рекурсивный спуск. Обычно указывается 1 или 2, и в зависимости от этого выбора получаются так называемые V- или W-циклы (см. рис. 4).

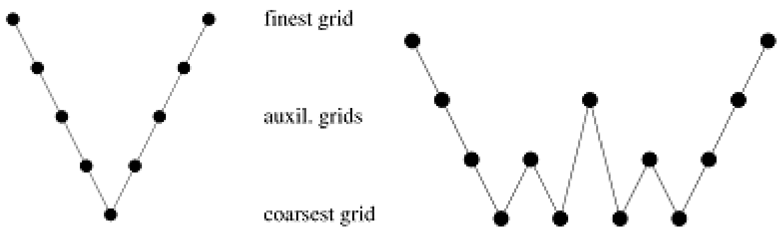

Рис. 4: V- и W-циклы

Алгоритм 1 Многосеточный метод

Многосеточный метод: $\mathrm{MGM}(\mathrm{K}_q, u_q, f_q, q)$

if $(q == 1)$ **then**

Решить $K_1 u_1 = f_1$ точно;

else

$d_q := f_q - k_q \cdot U_q;$ Вычисление невязки

$d_{q-1} := I_q^{q-1} \cdot q;$ Ограничение невязки на более грубую сетку (новая правая часть)

$\omega_{q-1}^0 := 0;$ Инициализация

$\omega_{q-1} := \mathsf{MGM}^\gamma(K_{q-1}, \omega_{q-1}^0, d_{q-1}, q-1);$ Решение системы с невязкой

$\omega_q := I_{q-1}^q \cdot \omega_{q-1};$ Интерполяция поправки

$u_q := u_q - k_q \cdot U_q;$ Вычисление невязки

fi

5. Метод декомпозиции области

Для достижения высокой степени параллелизма при решении эллиптических задач можно использовать метод декомпозиции области. Он заключается в разделении(декомпозиции) области эллиптической задачи на подобласти (см. рис 5), решении этой задачи на каждой подобласти параллельно и последующем объединении решений на подобластях. То есть полученная матрица жёсткости K представляется как

$$K = \begin{pmatrix} K_I & K_{I,E} & K_{I,V} \\ K_{E,I} & K_E & K_{E,V} \\ K_{V,I} & K_{V,E} & K_V \end{pmatrix}, \qquad (5.1)$$

где I – множество индексов узлов, попадающих внутрь подобластей, E – индексы узлов интерфейсной границы (узлов на гранях между подобластями), а V – индексы вершин (узлов на углах подобластей). Также под I_j имеются ввиду все узлы, принадлежащие j-й подобласти.

Ближе к середине прошлого века применение метода декомпозиции области было затруднено, так как дополнение Шура матрицы K_I в матрице жёсткости K было трудно вычислить. Но в в восьмидесятых годах М. Дрыей [5, 6] был рассмотрен предобуславливатель для решения отдельного блока подматрицы K_E, соответствующего одной грани

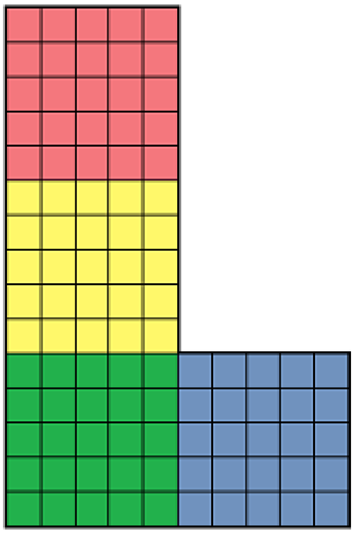

Рис. 5: Разделение на подобласти

интерфейсной границы. Он позволяет эффективно находить решение задачи на одной грани интерфейсной границы. Так как грани не имеют соседних точек (в терминах получившейся пятиточечной схемы), то дополнение Шура можно представить в виде блочно-диагональной матрицы и для каждого блока применить метод простой итерации с предобуславливателем.

Более подробное изложение метода декомпозиции области можно найти в соответствующей литературе. Он исследован во многих изданиях, например, в [1, 8], но пока существует не так много русскоязычных статей, содержащих описание реализации данного метода. В рамках данной работы был реализован метод декомпозиции области без перекрытия областей. Его обоснование можно найти в [8]. В алгоритме 2 описан предобуславливатель метода простой итерации для решения исходной задачи. Обозначения: J - число подобластей, на которые разбивается задача, v - вектор решения, состоящий из подвекторов, соответствующих подмножествам элементов (I, E, V), f - вектор правых частей, состоящий из подвекторов, соответствующих подмножествам элементов (I, E, V), Q - число граней, которые получаются в результате разбиения, $\mathcal{K}_{I_j}^{-1}$ – солвер локальной задачи Дирихле на подобласти, $\mathcal{P}_{I_j \leftarrow E_q}$ – оператор продолжения решения с граней на внутренние точки подобластей, $\mathcal{P}_{I_j \leftarrow E_q}^{T}$ – оператор продолжения решения с внутренних точек подобластей на грани.

Алгоритм 2 Метод декомпозиции области

Метод декомпозиции области
for $j = 1, 2, ..., J$ **do**
    $\mathsf{v}_{I_j} := \mathcal{K}_{I_j}^{-1} f_{I_j}$
    for q **such that** $E_{j,q} \in \partial\Omega_j$ **do**
        $\mathsf{f}_{E_{j,q}}^{(1)} := \mathcal{P}_{I_j \leftarrow E_q}^{T} \mathsf{f}_{I_j}$
    od
od
for $q = 1, 2, ..., Q$ **do**
    for $j = j$', j" **such that** $\partial\Omega_{j'} \cap \partial\Omega_{j''} = \overline{E_q}$ **do**
        $\mathsf{f}_{E_q} := f_{E_q} + f_{E_q}^{(1)}$
    od
    $\mathsf{v}_{E_q} := \mathcal{S}_{E_q}^{-1} \mathsf{f}_{E_q}$
od
for $j = 1, 2, ..., J$ **do**
    for q **such that** $E_{j,q} \in \partial\Omega_j$ **do**
        $\mathsf{v}_{I_{j,q}}^{(1)} := \mathcal{P}_{I_j \leftarrow E_q} \mathsf{v}_{E_q}$
        $\mathsf{v}_{I_j} := \mathsf{v}_{I_j} + \mathsf{v}_{I_{j,q}}^{(1)}$
    od
od

Предобуславливатель используется только для решения задачи на внутренних элементах и элементах интерфейсной границы. Для решения на вершинах используется точный солвер $\mathcal{K}_{V}^{-1}$, использующий метод Гаусса. Для вершин можно построить отдельную матрицу жёсткости, так как задача для вершин эквивалентна исходной задаче, только на грубой сетке, и точно решить систему алгебраических линейных уравнений.

6. Реализация метода декомпозиции области

В рамках данной работы была написана параллельная программа, реализующая метод декомпозиции области. Программа написана на языке C++ с использованием OpenMP [4], реализованной в компиляторе Intel Cluster Studio 2013 [11].

В программе описан класс Triangulation, в объектах которого хранятся данные о триангуляции данной области. На основе экземпляра класса Triangulation строится экземпляр класса Decomposition, в котором содержатся данные о разбиение области на подобласти, на которых уравнение Пуассона будет решатся параллельно. Также в экземплярах класса Decomposition содержится информация о гранях и вершинах декомпозиции, которые обрабатываются особым образом, описанным выше.

В системе доступен выбор таких солверов для подобластей, как метод Гаусса и многосеточный метод. Последний был реализован во внешней библиотеке AMGCL [3]. Для решения предобуславливателя Дрыйи [5, 6] использовался метод преобразования Фурье, реализованный в библиотеке Fastest Fourier Transform in the West [7].

Программа запускалась на L-образной области (см. рис. 1) с параметрами N и P, где N - число степеней свободы на кратчайшей стороне фигуры, а P - число подбластей, на которые делится наша область. Замеры времени работы программы проводились на кластере
Т-Платформы вычислительного центра СПбГУ [10]. В кластере содержится 48 вычислительных узлов Dexus, в каждом из которых имеется 2 CPU Intel Xeon Processor E5335 2.0ГГц, 16 Гб RAM. Они связаны сетью InfiniBand 20Gb. Результаты измерений времени выполнения приведены в табл. 2 и на рис. 6.

По этим данным становится понятно, что реализованный метод позволяет эффективно решать задачу Пуассона для данной области приведены большом количестве процессоров. Эта таблица соответствуют данным, полученным Формаджей и другими в статье сборника [1].

N/P	4	16	64	256
64	1.37552	0.4864	0.27790	0.24750
128	2.28633	0.93803	0.59279	0.52801
256	7.97373	4.11267	2.59901	2.31497
512	36.11017	23.04824	14.56534	12.97354
1024	145.01013	113.12449	71.48904	63.67624
2048	570.07496	538.11567	340.06229	302.89802

Таблица 2: Сравнение времени выполнения (в секундах) программы при различных входных параметрах

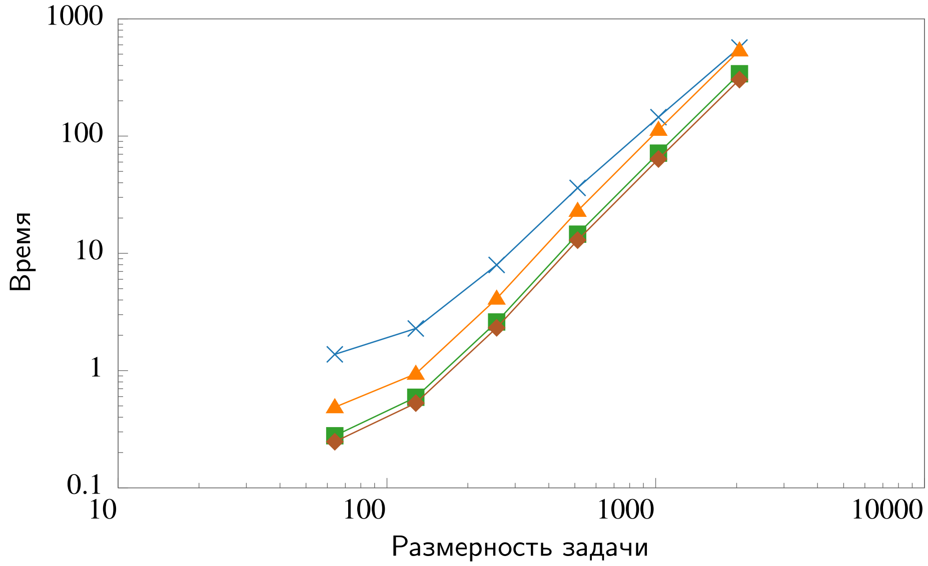

Рис. 6: Зависимость времени вычисления от размеров системы

Также из таблицы следует, что при наличии многопроцессорной системы для решения эллиптических задач полезно использовать метод декомпозиции области с большим числом подобластей.

Заключение

В рамках данной работы был рассмотрен метод конечных элементов. Для решения получившейся системы использовался метод декомпозиции области и многосеточный метод. Была реализована параллельная программа и запущена на вычислительном кластере.

Рассмотренный алгоритм позволяет быстро решать уравнение Пуассона. Также он показывает значительное ускорение при увеличении числа подобластей, что доказывает возможность его эффективной параллелизации и высокую гранулярность. Использование метода декомпозиции области может эффективно использоваться для быстрого численного решения эллиптических задач на параллельных компьютерах и кластерах.

Список литературы

[1] Are Magnus Bruaset, Aslak Tveito. Numerical Solution of Partial Differential Equations on Parallel Computers. — Springer Berlin Heidelberg, 2006. — ISBN: 978-3-540-31619-0.

[2] Craig C. Douglas, Gundolf Haase, Ulrich Langer. A Tutorial on Elliptic PDE Solvers and their Parallelization. — Society for Industrial & Applied Mathematics, 2002. — ISBN: 978-0-89871-541-5.

[3] Denis Demidov. AMGCL // Github. — 2012. — URL: `https://github.com/ddemidov/amgcl` (online; accessed: 10.05.2017).

[4] Home - OpenMP. — URL: `http://www.openmp.org/`.

[5] M. Dryja. A Capacitance Matrix Method for Dirichlet Problem on Polygon Region // Numerische Mathematik. — 1982. — Vol. 39. — P. 51–64.

[6] M. Dryja. A finite element-capacitance method for elliptic problems on regions partitioned into subregions // Numerische Mathematik. — 1984. — Vol. 44. — P. 153–168.

[7] Matteo Frigo, Steven G. Johnson. FFTW Home Page. — 1997. — URL: `http://www.fftw.org/` (online; accessed: 10.05.2017).

[8] Vadim Glebovich Korneev, Ulrich Langer. Dirichlet-Dirichlet domain decomposition methods for elliptic problems. — World Scientific, 2015. — ISBN: 978-981-4578-45-5. — World Scientific : http://www.worldscientific.com/worldscibooks/10.1142/9035.

[9] Даутов Р.З., Карчевский М.М. Введение в теорию метода конечных элементов. — КГУ, 2004. — ISBN: 978-5-98180-993-4.

[10] Исследования были проведены с использованием вычислительных ресурсов Ресурсного Центра "Вычислительный центр СПбГУ". — URL: `http://cc.spbu.ru/`.

[11] Официальный сайт Intel Cluster Studio. — URL: http://software.intel.com/en-us/intel-cluster-studio-xe.